

# Spatial and Temporal Interpolation of Multi-View Image Sequences

Tobias Gurdan\* \*\*, Martin R. Oswald\*, Daniel Gurdan\*\*, Daniel Cremers\*

\* Department of Computer Science, Technische Universität München

\*\* Ascending Technologies GmbH, Krailling, Germany

# Introduction

# Introduction

- Related work

# Introduction

- Related work
  - Beier and Neeley [BN92]

# Introduction

- Related work



# Introduction

- Related work
  - Beier and Neeley [BN92]
  - Seitz and Dyer [SD95, SD96]
  - Zitnick et al. [ZKU+04]

# Introduction

- Related work

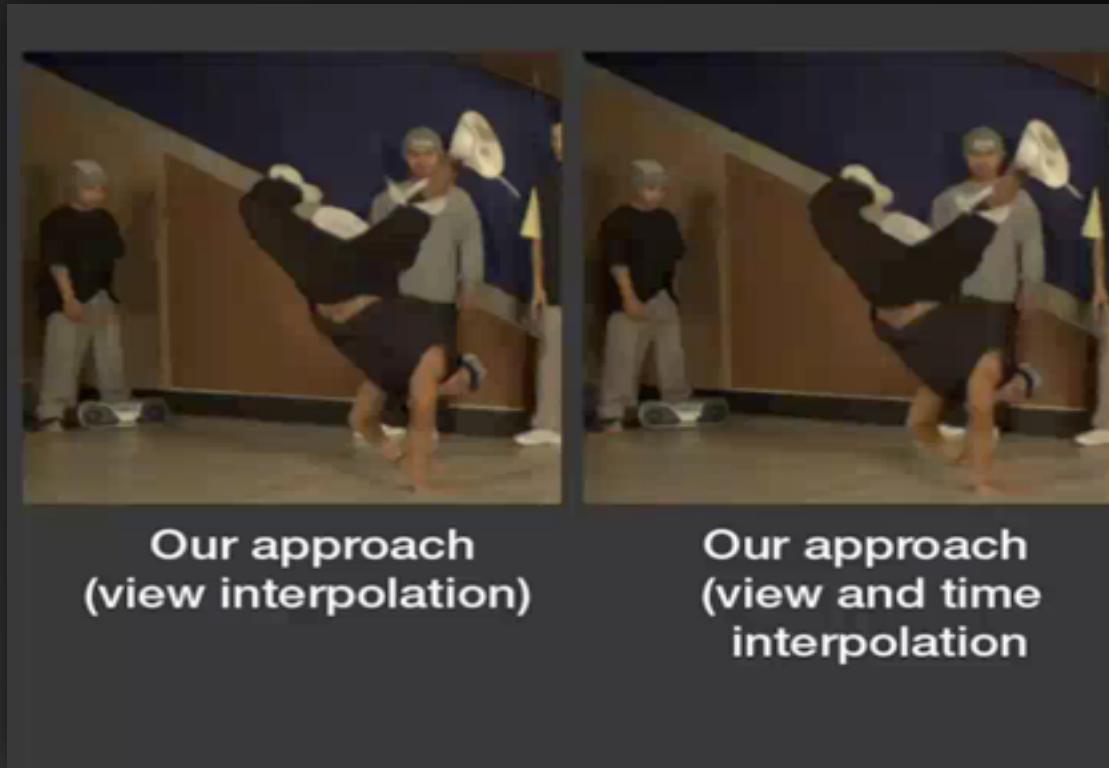


# Introduction

- Related work
  - Beier and Neeley [BN92]
  - Seitz and Dyer [SD95, SD96]
  - Zitnick et al. [ZKU+04]
  - Lipski et al. [LLBM09, Lipski13]

# Introduction

- Related work



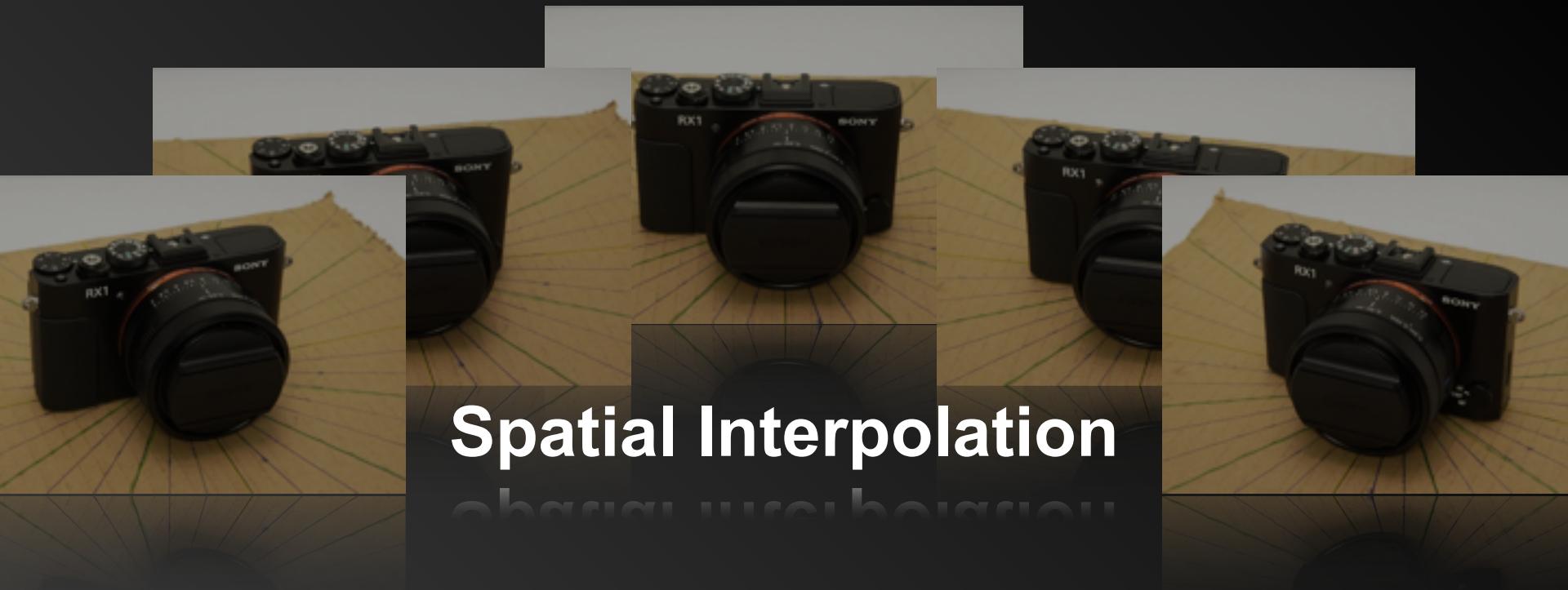
# Introduction

- Motivation of this work



# Introduction

- Goals of this work
  - Spatial and temporal interpolation
  - Easy to use
  - Autonomous
  - No restriction on setup
  - Wide baseline



# Spatial Interpolation

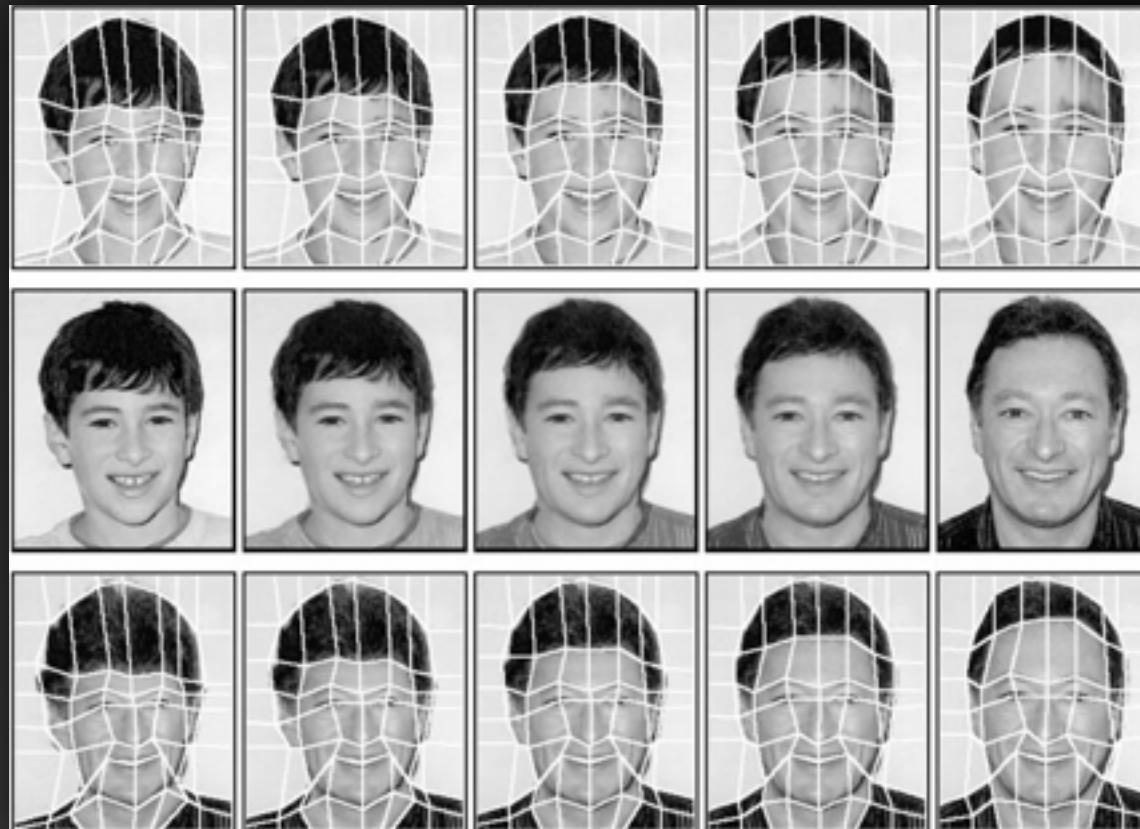
obliqua uice ubiqui

# Spatial Interpolation

- Sparse approach
  1. Find sparse set of features
  2. Match features
  3. Compute textured triangle meshes
  4. Linearly interpolate vertex positions
  5. Blend warped images

# Spatial Interpolation

- Example: Face Morphing [Wol98]



# Spatial Interpolation

Matcher

# Matcher

- Robust Matcher

# Matcher

- Robust Matcher
  - Source and target image



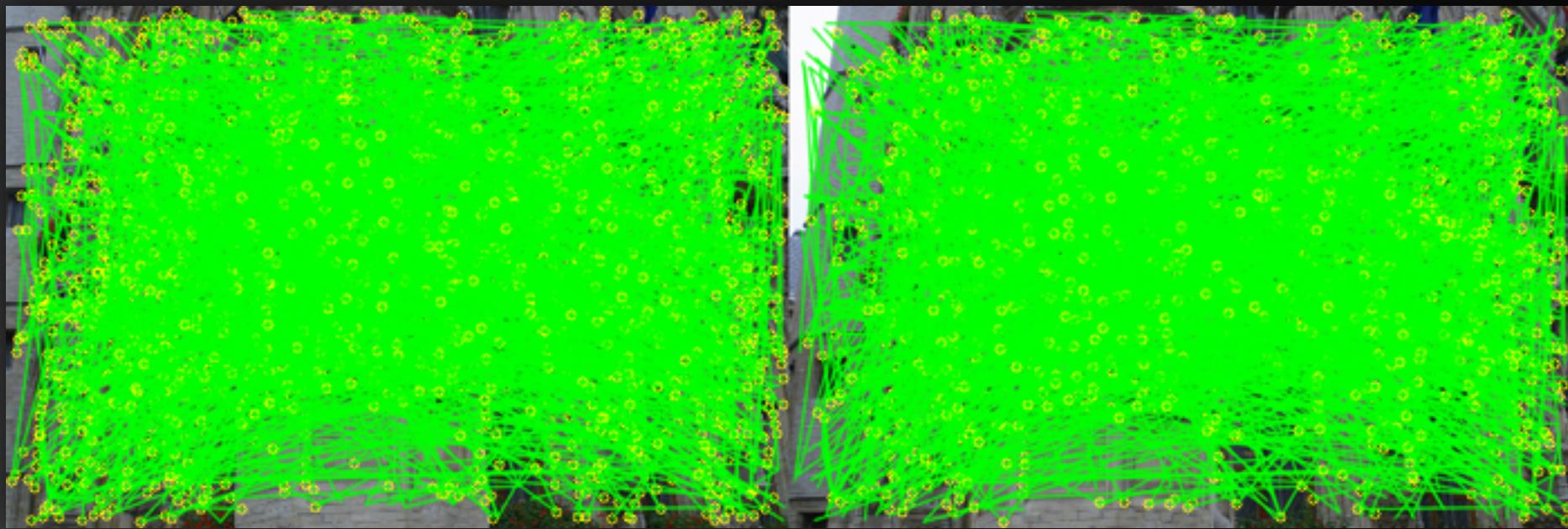
# Matcher

- Robust Matcher
  - SIFT features



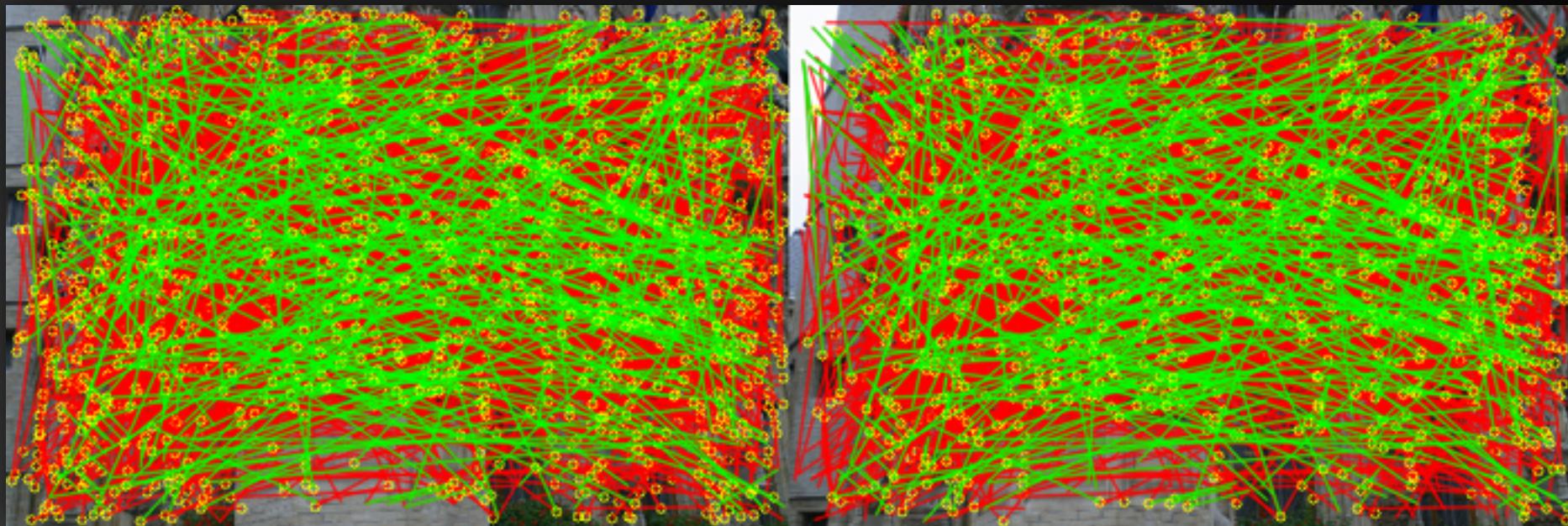
# Matcher

- Robust Matcher
  - SIFT descriptor matching



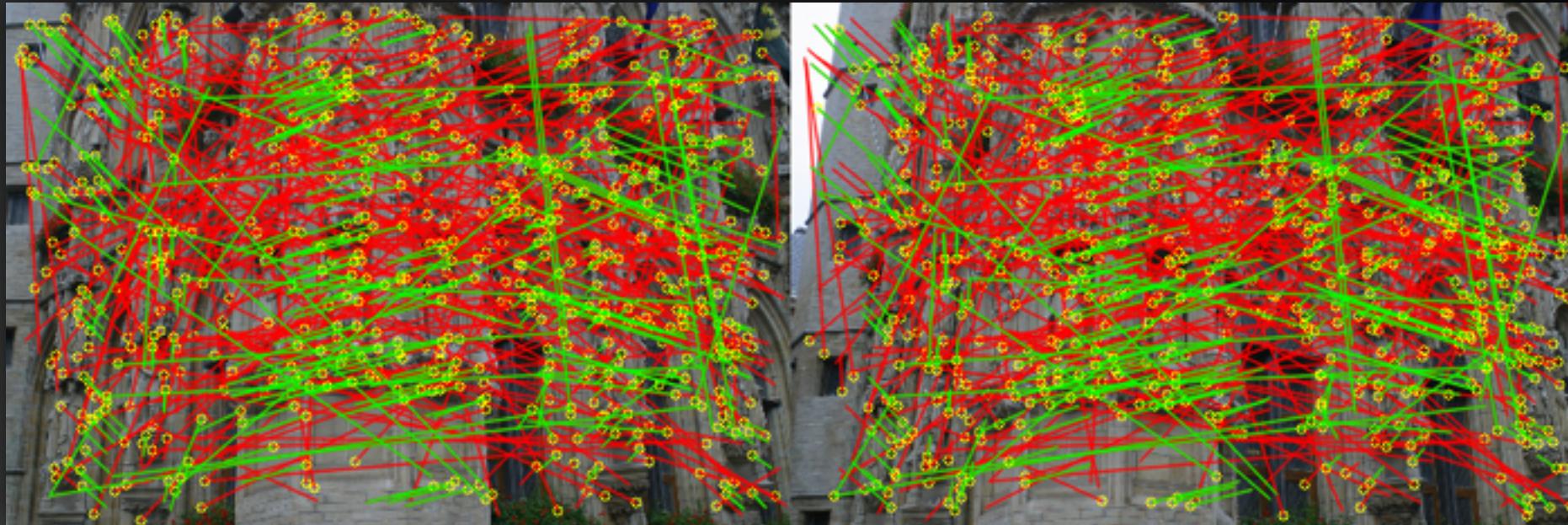
# Matcher

- Robust Matcher
  - Symmetric matches



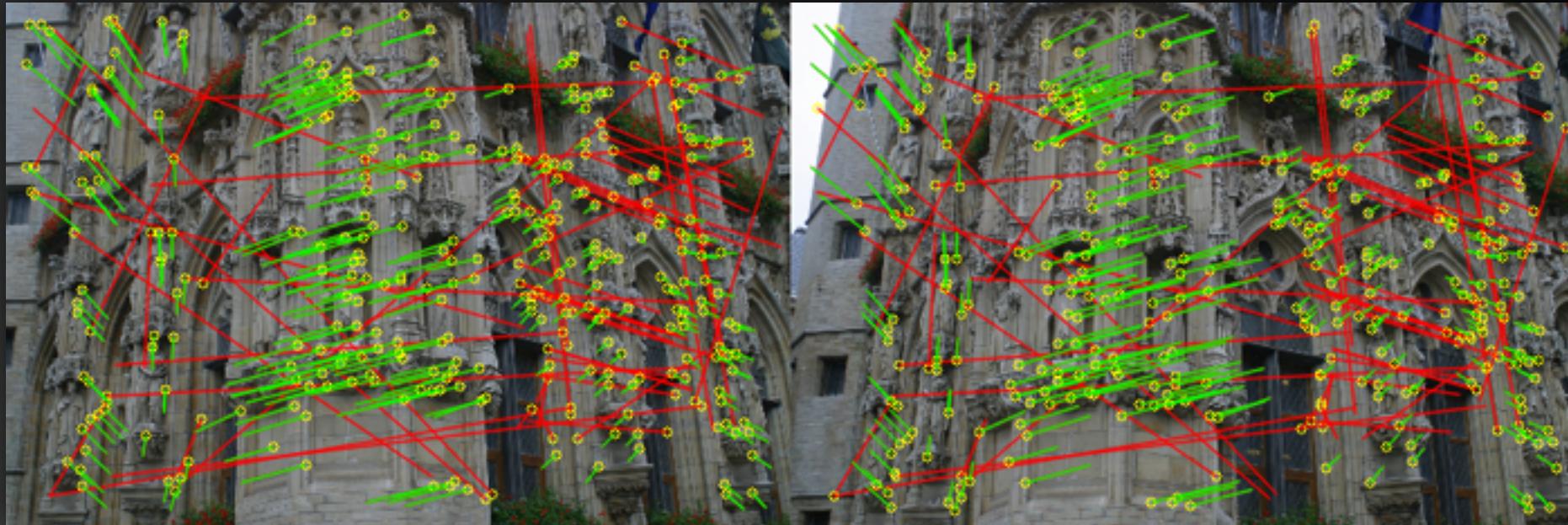
# Matcher

- Robust Matcher
  - Thresholded matches



# Matcher

- Robust Matcher
  - RANSAC filtering



# Matcher

- Robust Matcher
  - Final matching



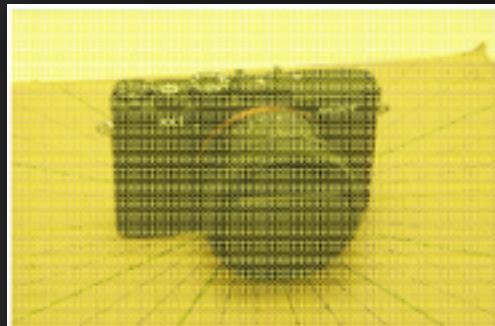
# Matcher

- Robust Matcher
  - Matching is done in space of all descriptors
  - SIFT detector not suitable for warping

- **Distribution** - Not too close to get good meshes, not too far apart which might lead to missed regions
- **Coverage** - Include most to all significant points, e.g. object outlines, corners, foreground / background
- **COASTLINE** - ~~THE COASTLINE OF THE RIVER DANUBE IS COASTLINE?~~

# Matcher

- Feature Detectors



(a) Dense features



(b) FAST features



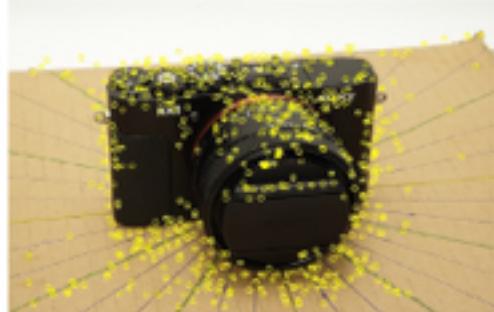
(c) GFTT features



(d) ORB features



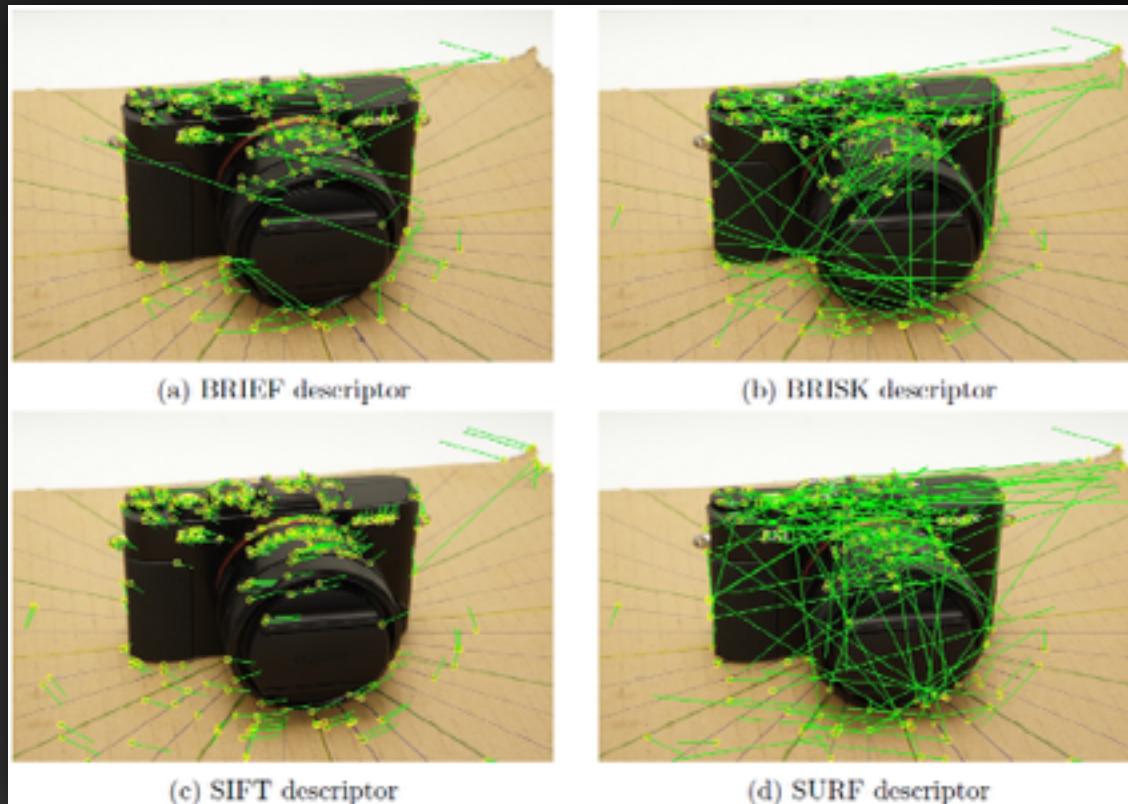
(e) SIFT features



(f) SURF features

# Matcher

- Feature Descriptors



# Matcher

- Robust Matcher
  - Matching is done in space of all features
  - SIFT features not suitable for warping
- Our approach
  - Use more versatile features (GFTT + SIFT)
  - Constrain search space using epipolar geometry as guide
  - ASIFT as fallback

# Matcher

- Epipolar Guided Matcher
  - Source and target image



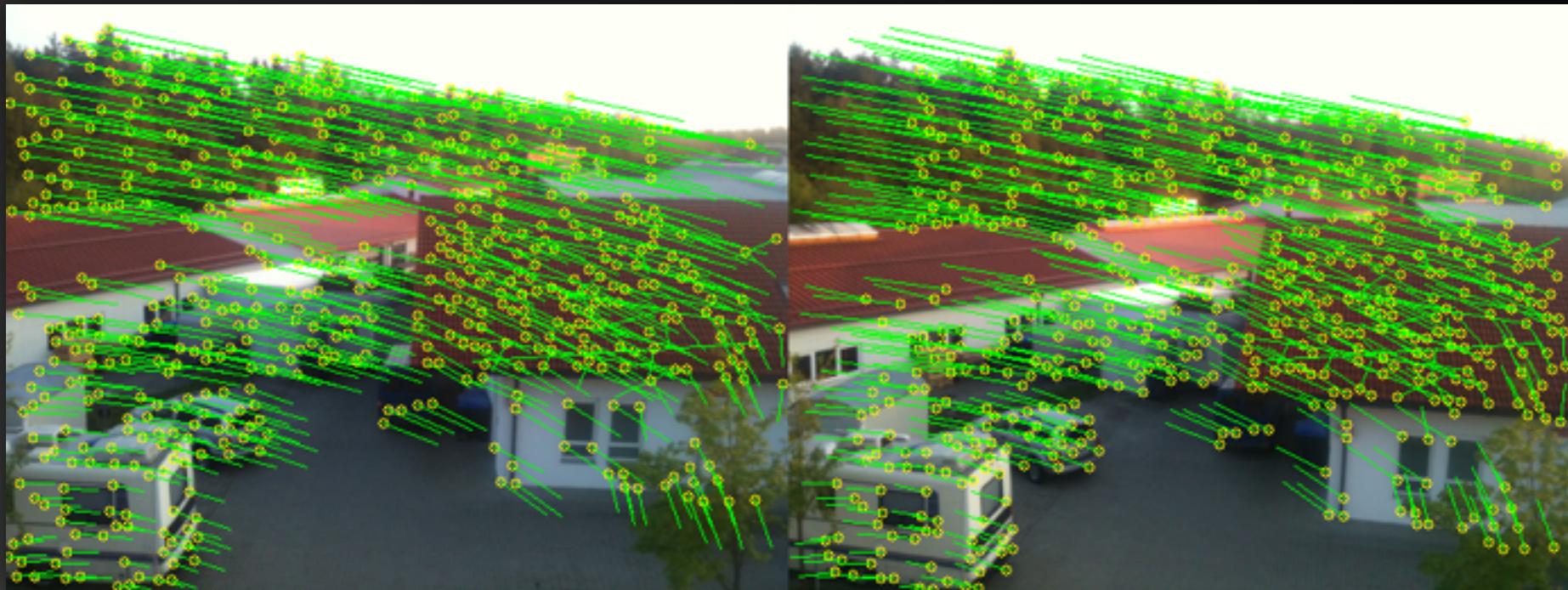
# Matcher

- Epipolar Guided Matcher
  - Robust Matcher result



# Matcher

- Epipolar Guided Matcher
  - Guided matching using epipolar constraint

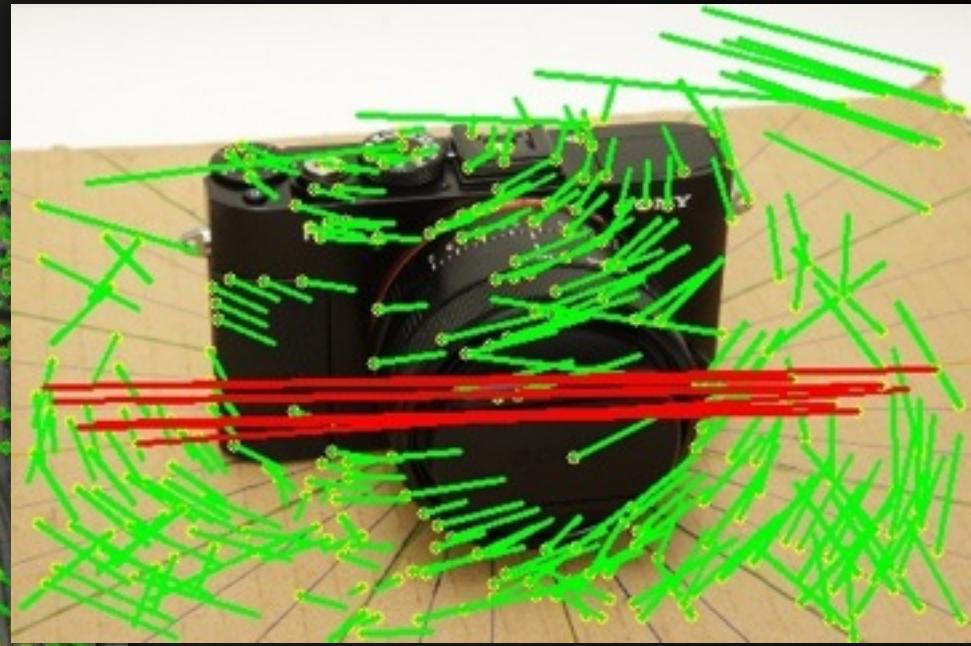
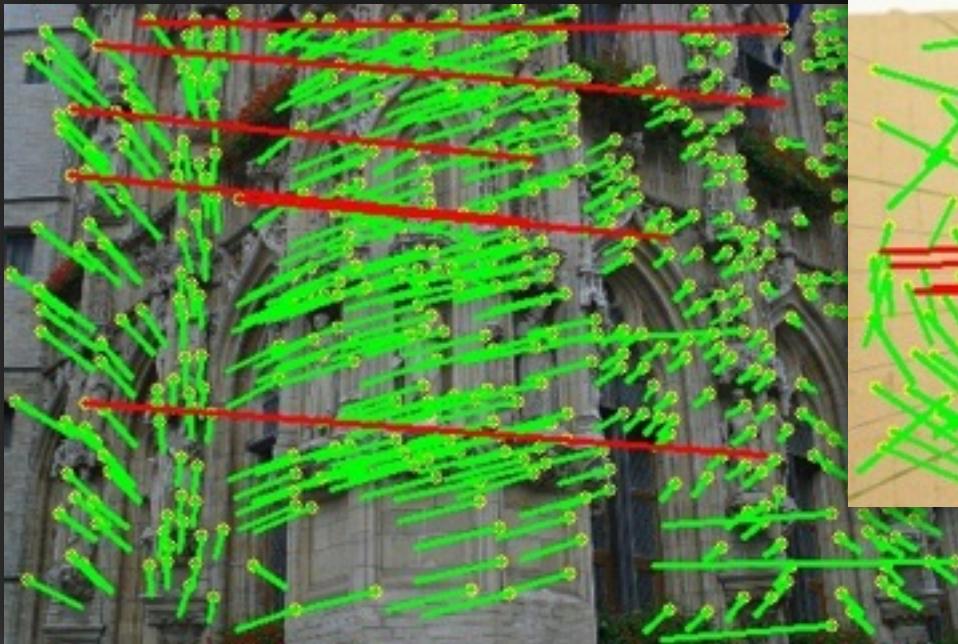


# Filter

- Playground Filter
  - Global filtering
  - Partition matches into three sets
  - Corresponds to simplified motion model
  - Discard set with largest displacement ...
  - ... if the distance is above a certain threshold
  - Efficiently solvable using k-Means clustering

# Filter

- Playground Filter

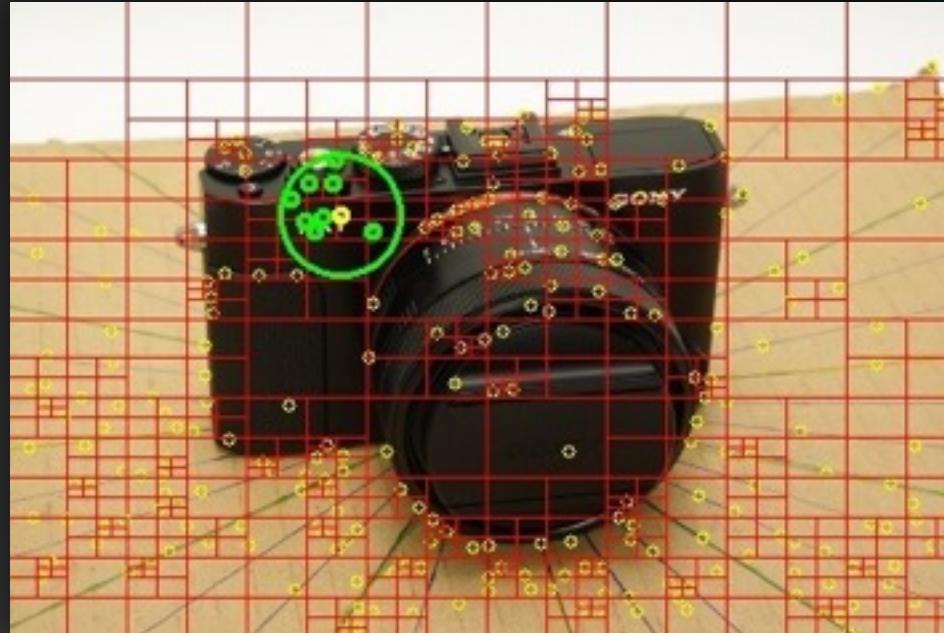


# Filter

- Playground Filter
- Council Filter
  - Local filter
  - Look at small neighbourhoods
  - Each keypoint in that region votes for a candidate
  - Accept candidate as inlier only if it receives enough votes

# Filter

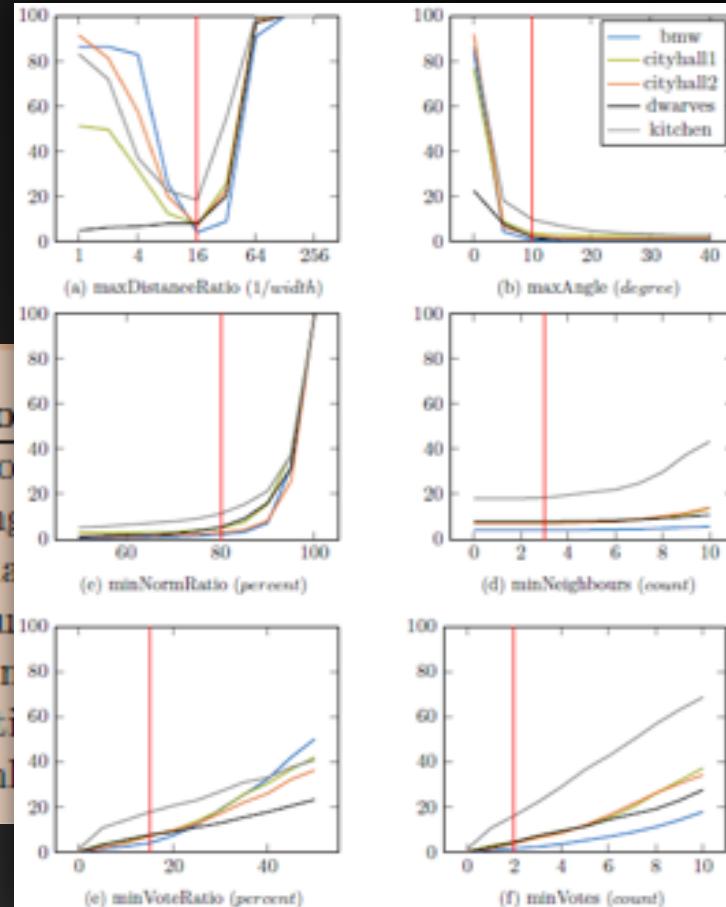
- Playground Filter
- Council Filter



# Filter

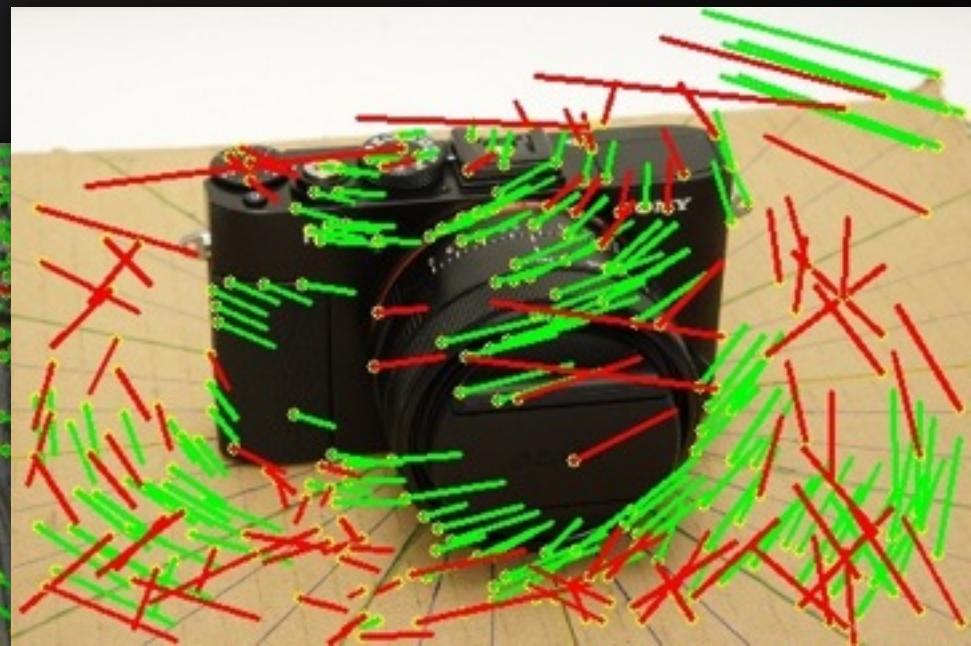
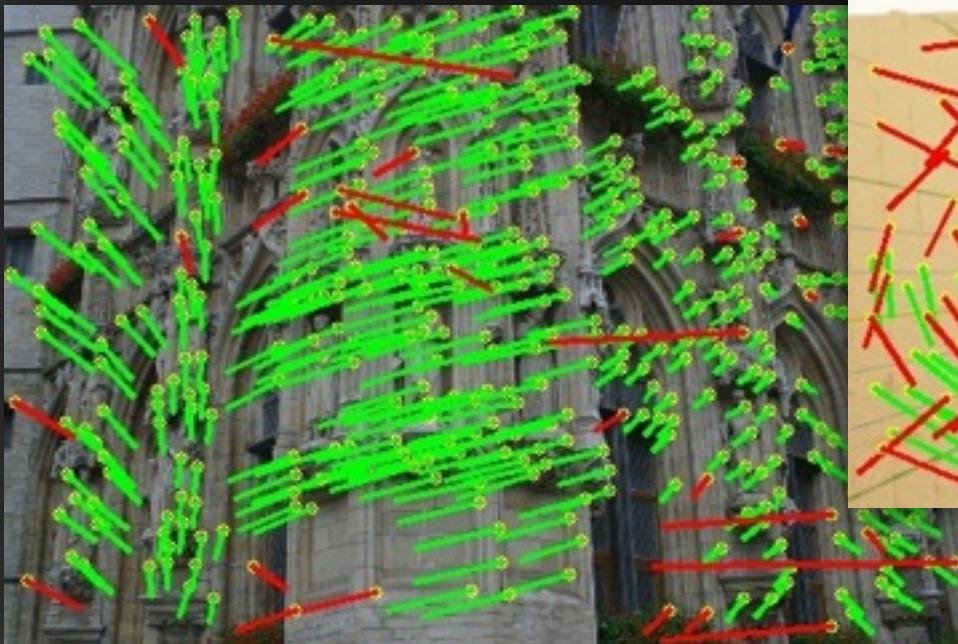
- Playground Filter
- Council Filter

Variable	Value	Explanation
maxDistanceRatio	1/16	neighbourhood
maxAngle	10	maximal angle
minNormRatio	0.8	minimal relative
minNeighbours	3	minimal count
minVotes	2	minimal number
minVoteRatio	0.15	minimal ratio
nonQuorumPass	false	accept as invalid



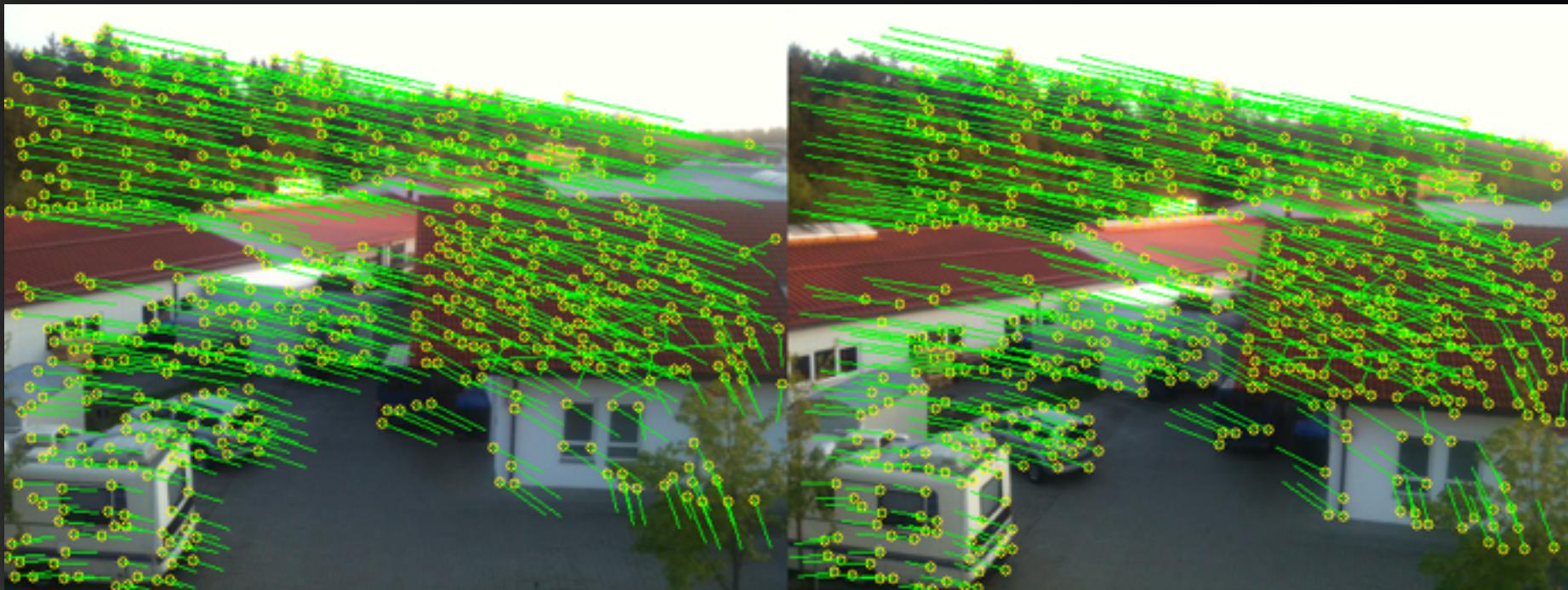
# Filter

- Playground Filter
- Council Filter



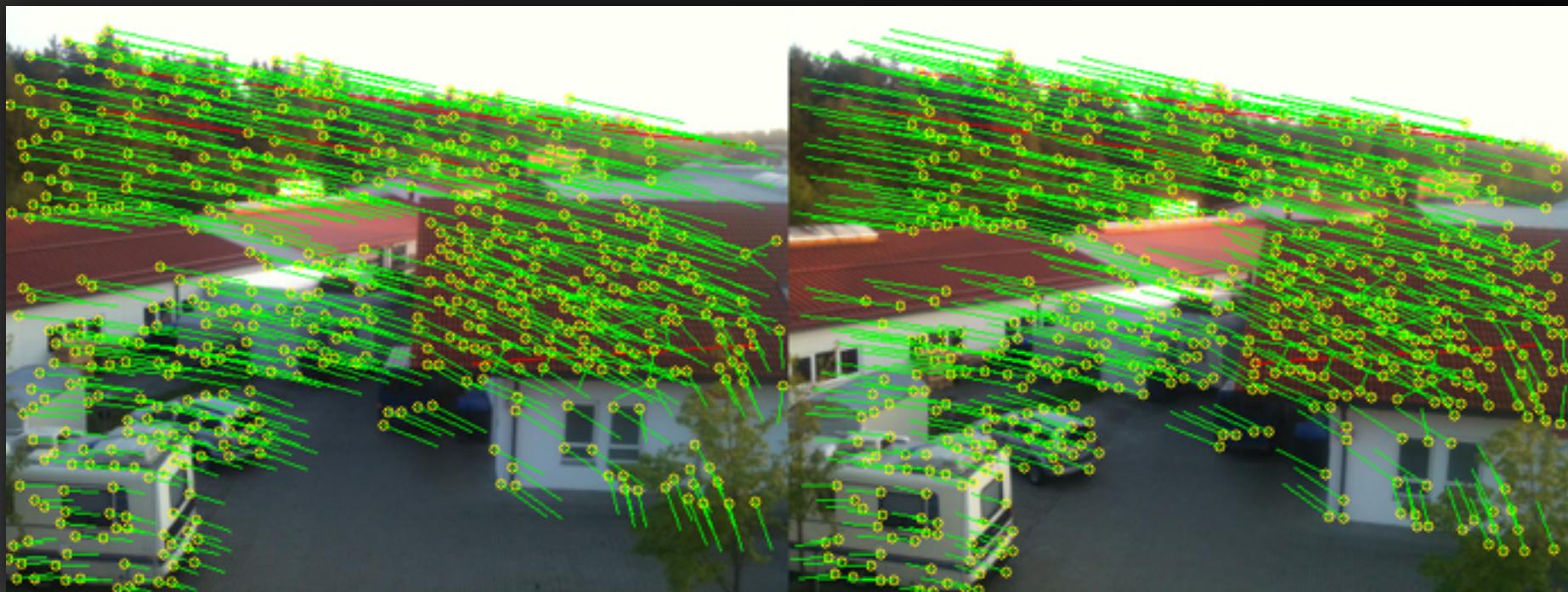
# Filter

- Epipolar Guided Matcher (cont.)
  - Guided matching using epipolar constraint



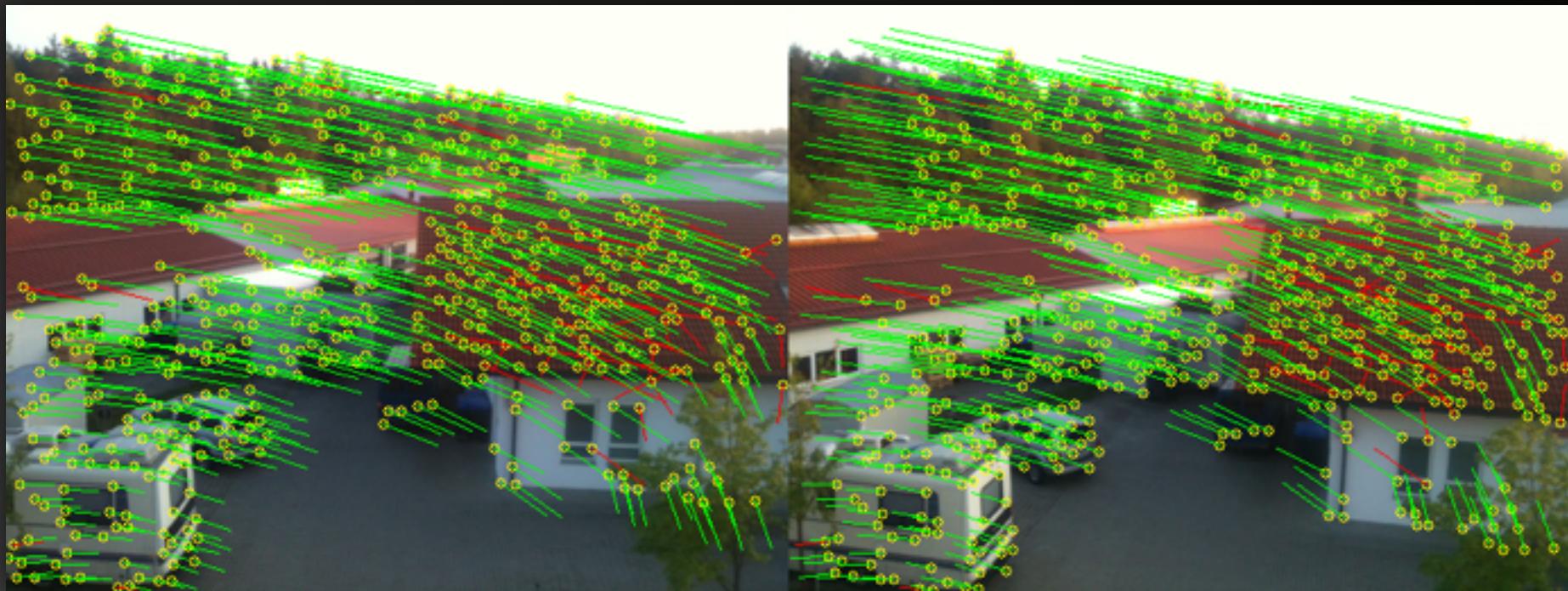
# Filter

- Epipolar Guided Matcher (cont.)
  - Apply Playground Filter



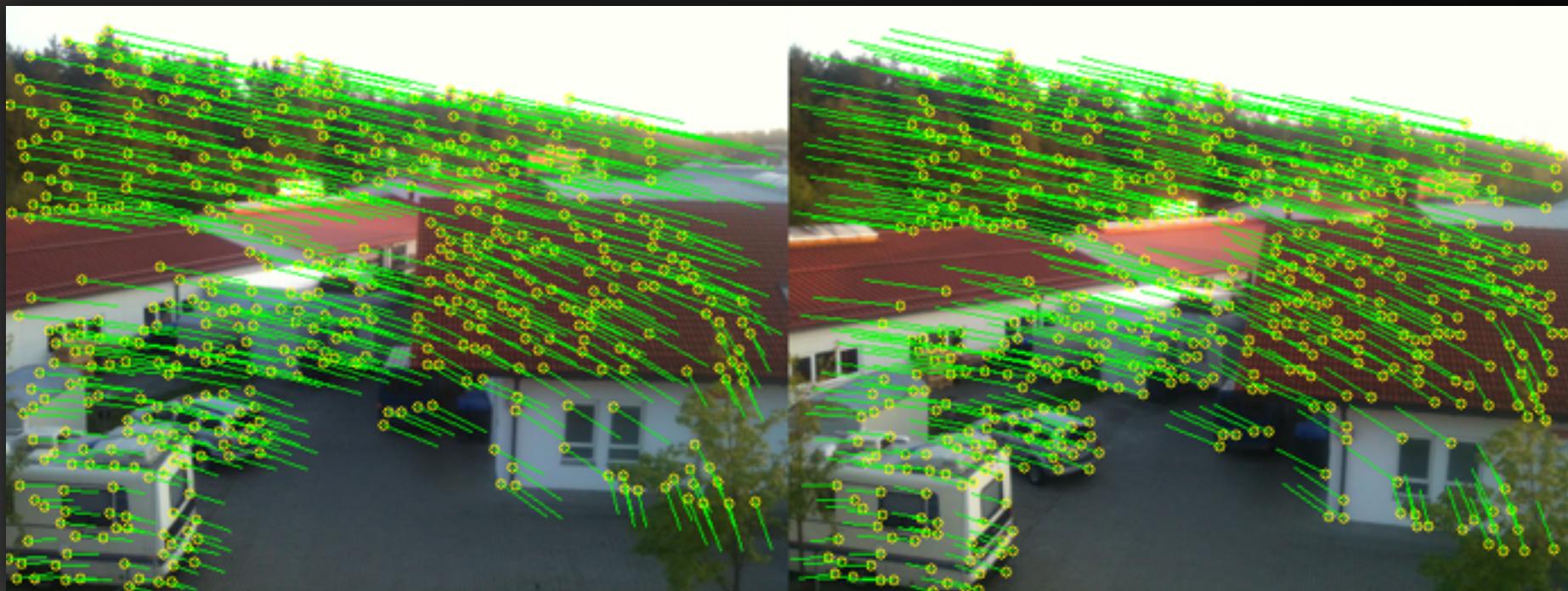
# Filter

- Epipolar Guided Matcher (cont.)
  - Apply Council Filter



# Filter

- Epipolar Guided Matcher (cont.)
  - Final matching

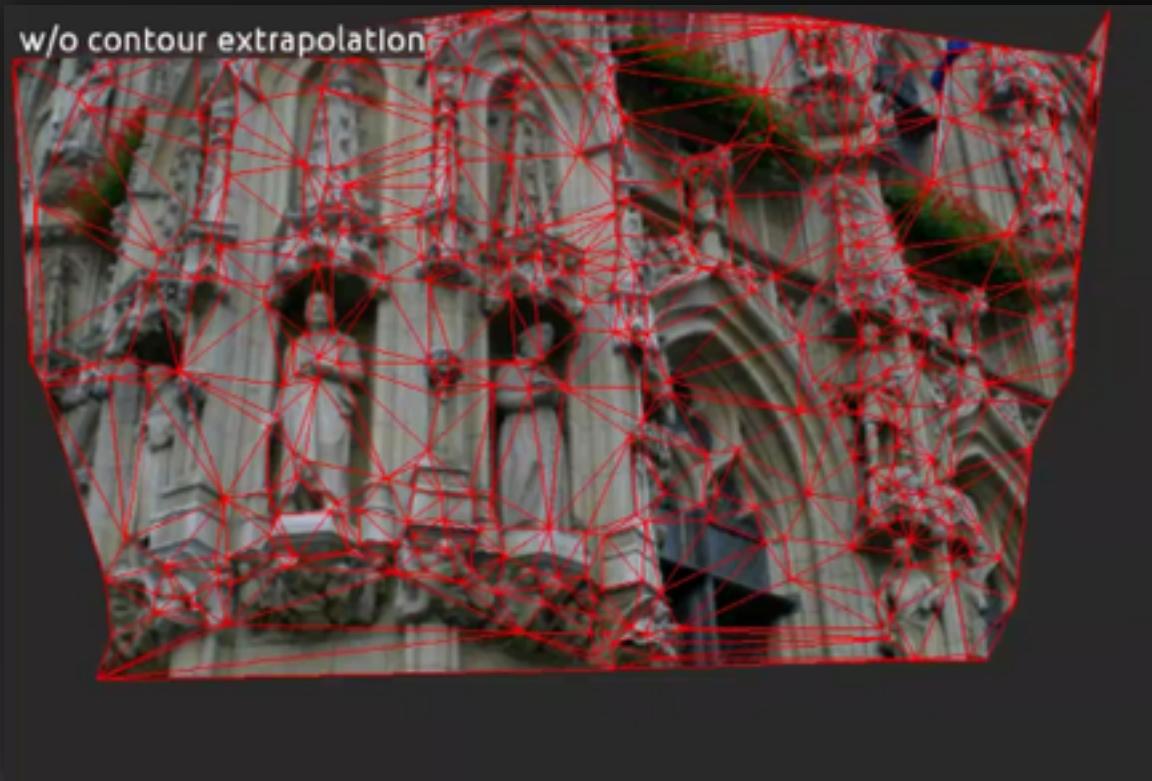


# Spatial Interpolation

Triangulation

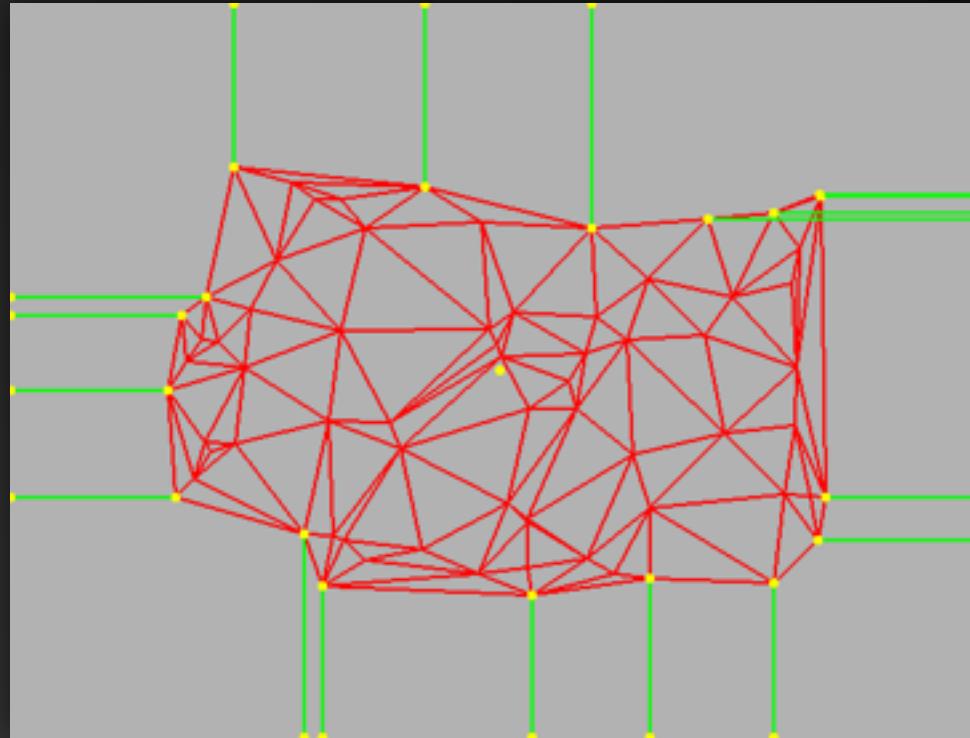
# Triangulation

- Delaunay Triangulation



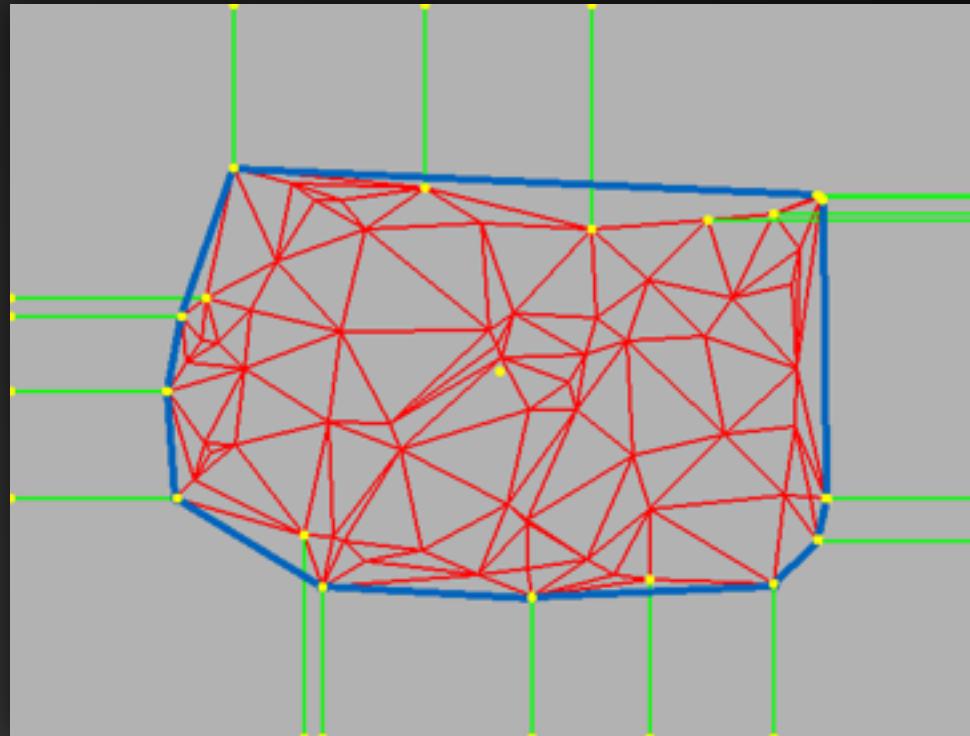
# Triangulation

- Border Extrapolation
  - Orthogonal projection



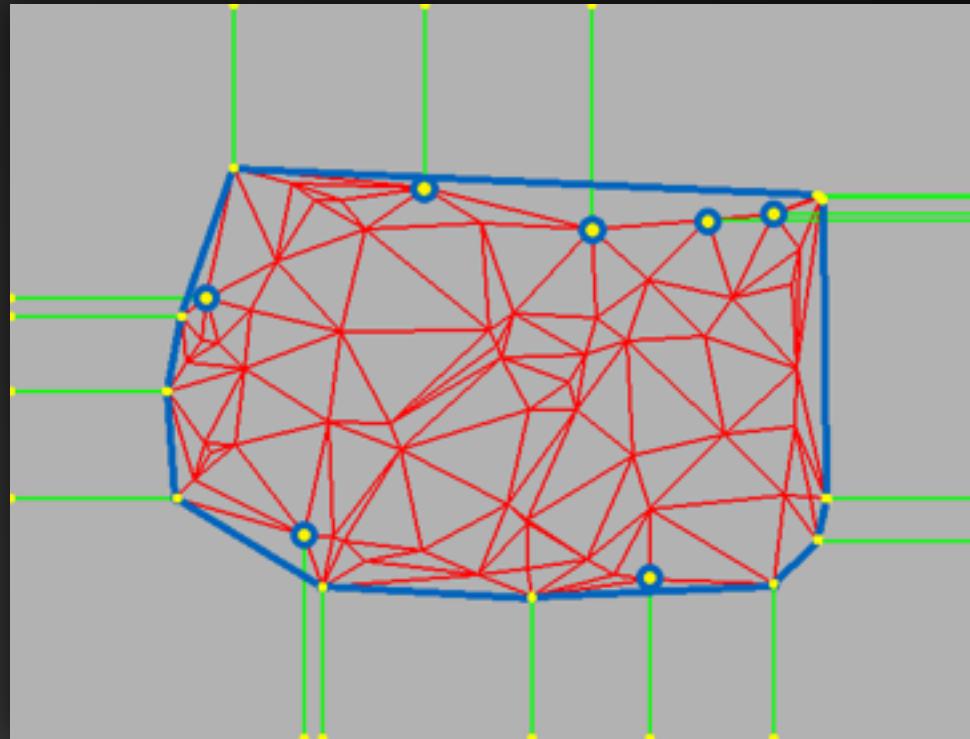
# Triangulation

- Border Extrapolation
  - Convex hull



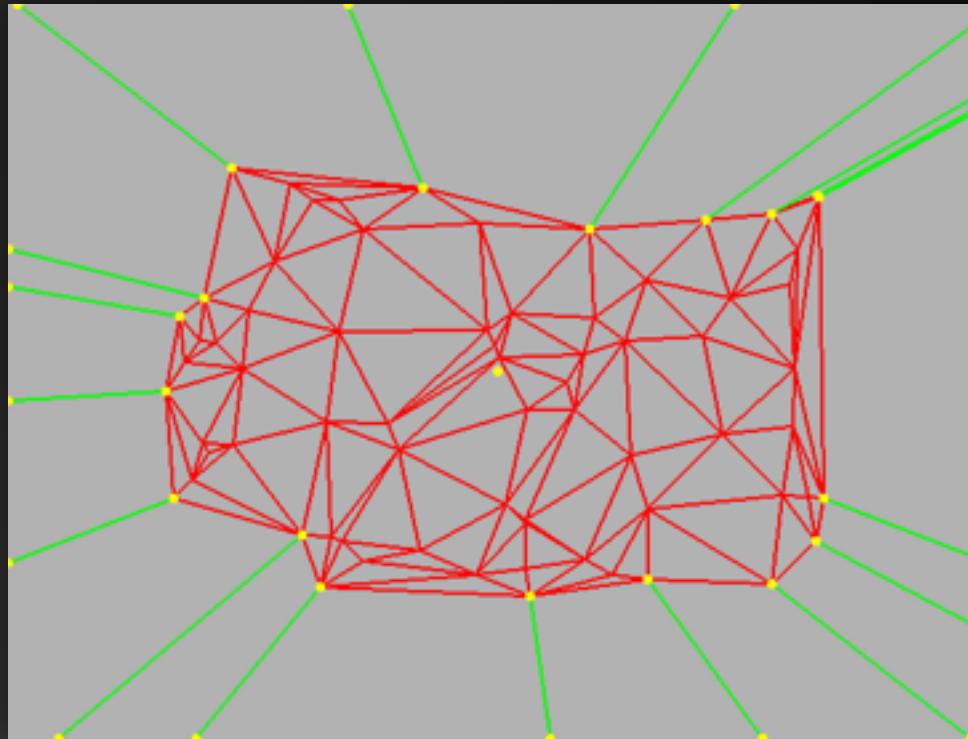
# Triangulation

- Border Extrapolation
  - Left out points



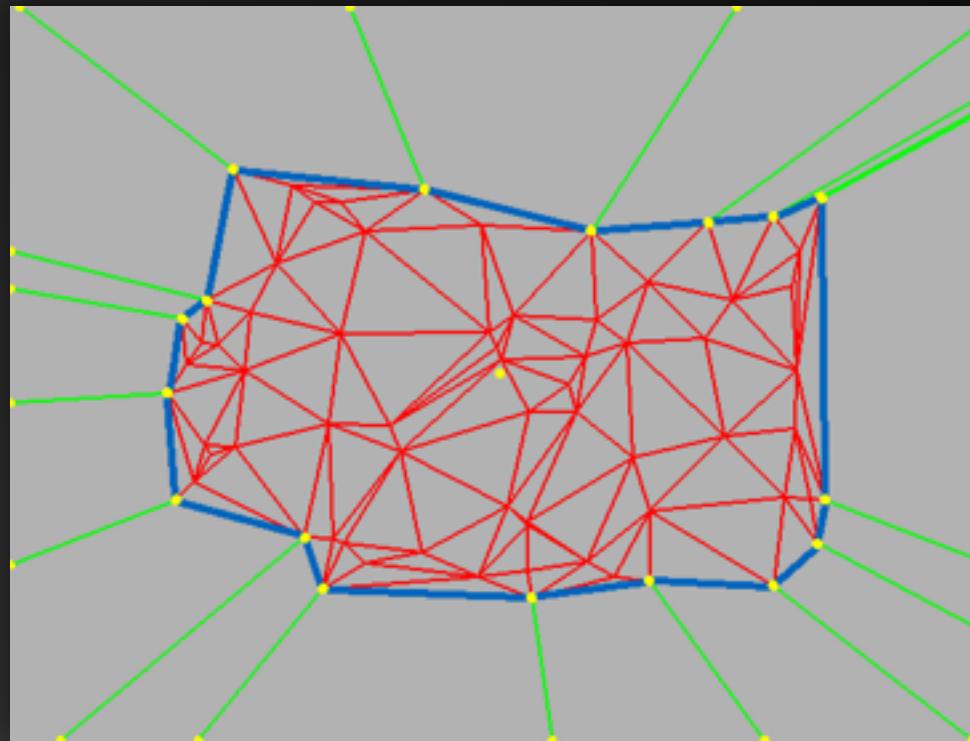
# Triangulation

- Border Extrapolation
  - Radial projection



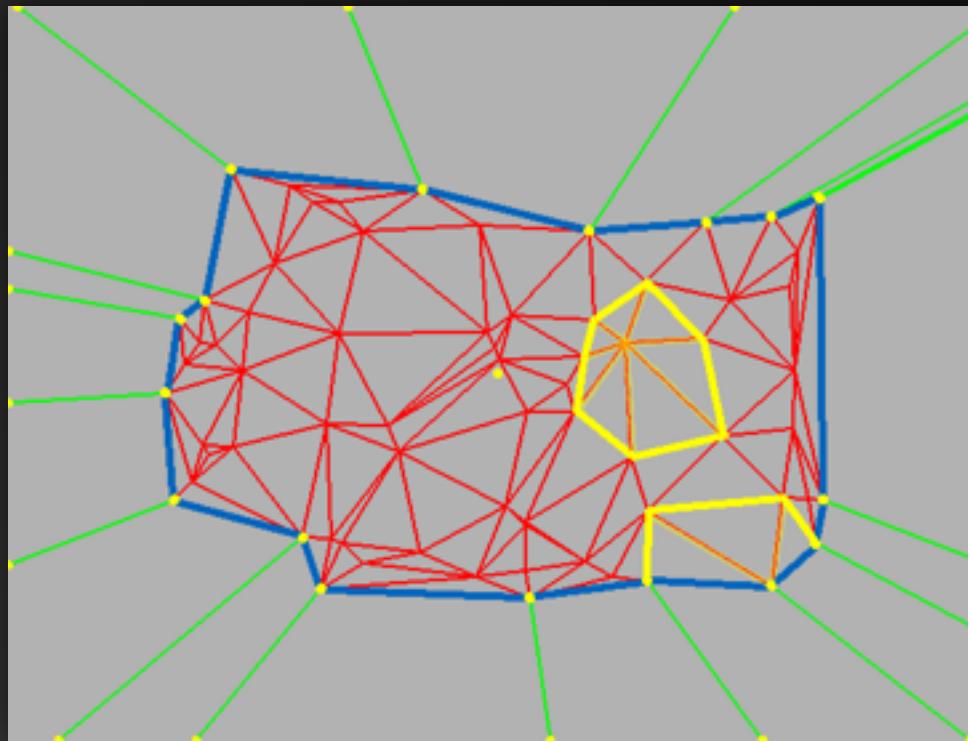
# Triangulation

- Border Extrapolation
  - Point set contour



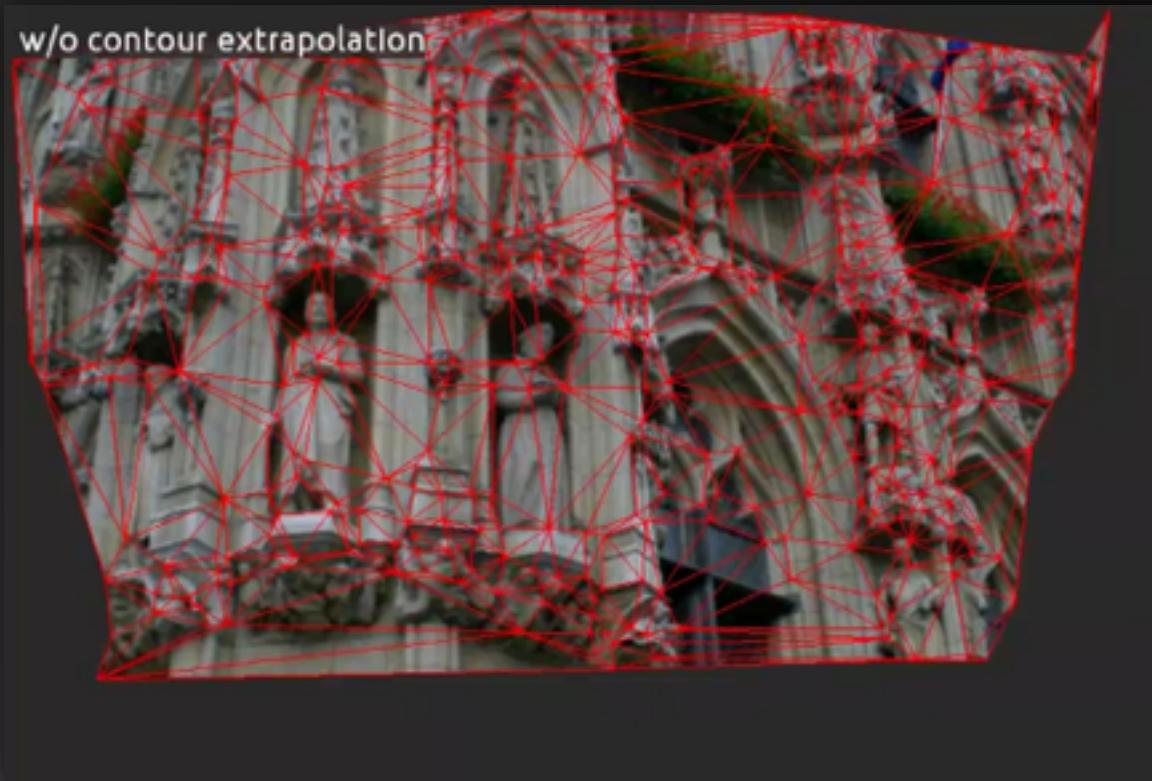
# Triangulation

- Border Extrapolation
  - Exemplary induced wheel graphs



# Triangulation

- Delaunay Triangulation



# Triangulation

- Delaunay Triangulation w. Border Extrapolation



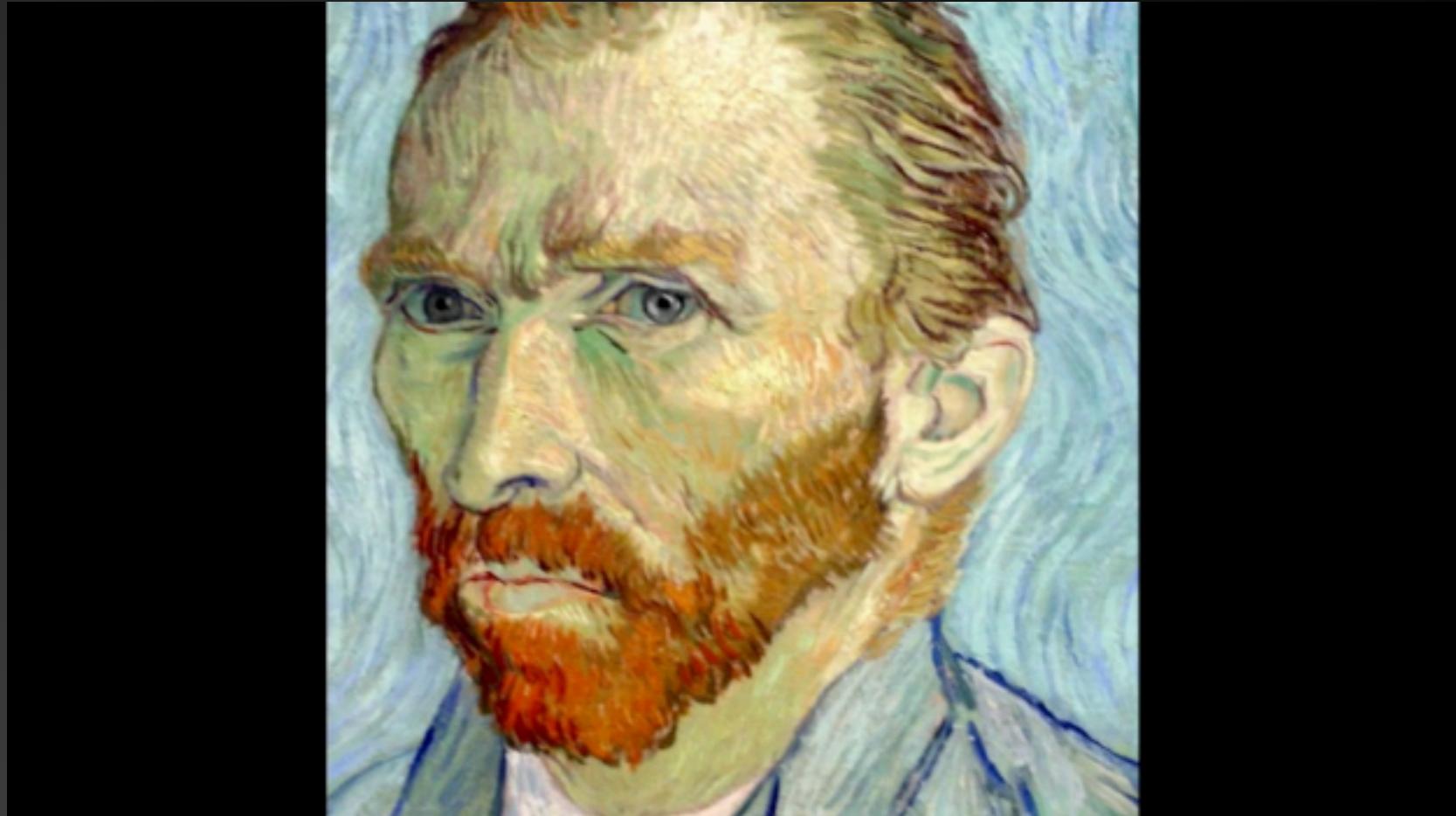
# Spatial Interpolation

View Morphing

# View Morphing

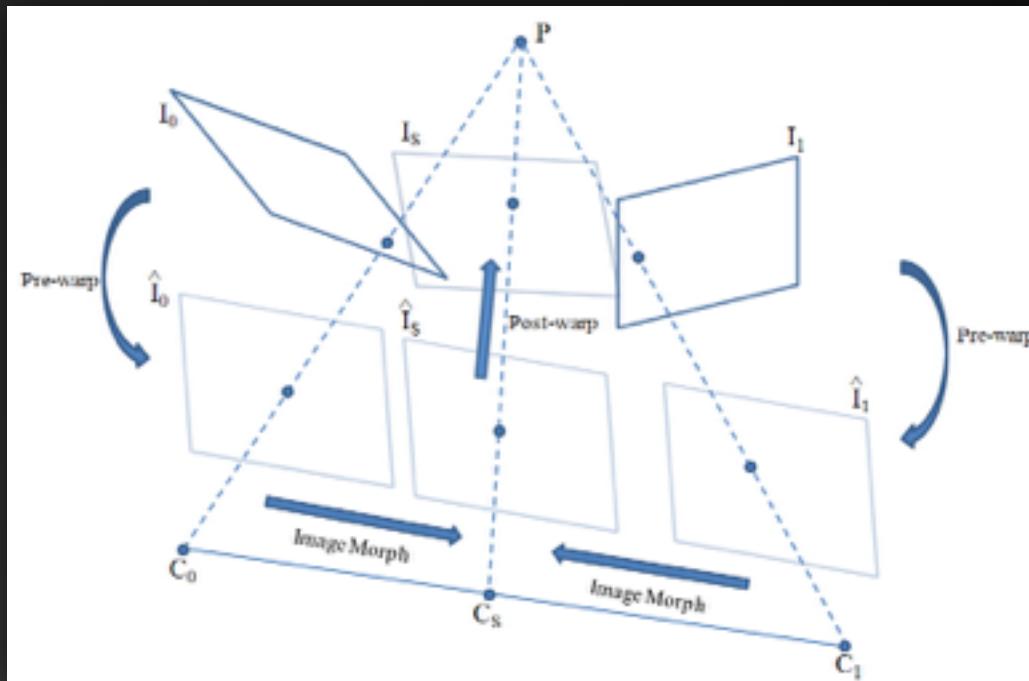
- Render textured triangle mesh
- Linearly interpolate triangle vertices from source to target
- Can result in unrealistic transformations!

# View Morphing



# View Morphing

- Seitz and Dyer introduce View Morphing [SD96]
- Consists of pre- and postwarping



# View Morphing

- Seitz and Dyer introduce View Morphing [SD96]
- Consists of pre- and postwarping
- Backprojection using homography interpolation, as introduced by Fragneto et al. [FFR+12]

Given two rectification homographies  $\mathbf{H}_1, \mathbf{H}_2$ . An intermediate homography at  $t \in [0, 1]$  is given by

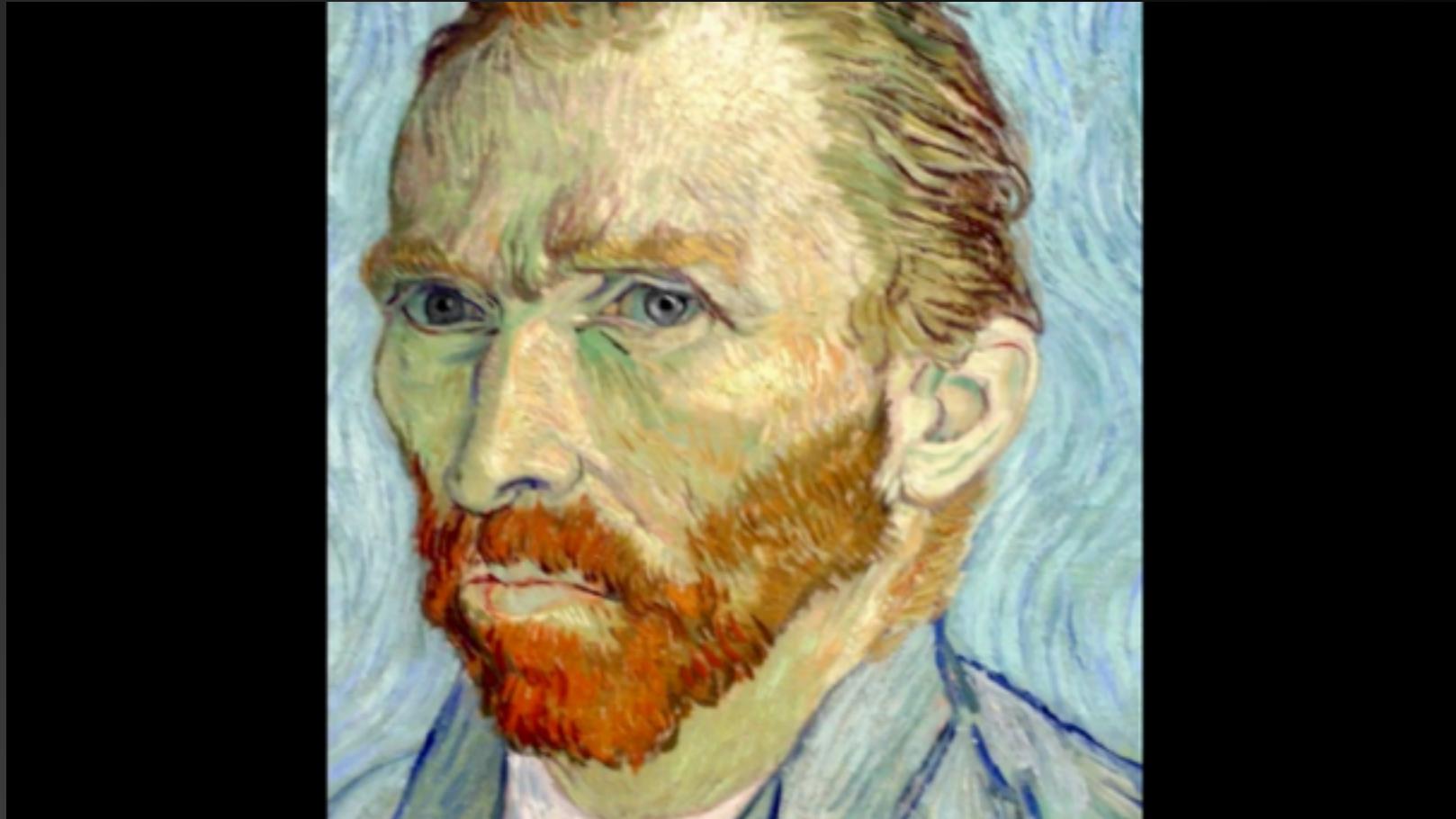
$$\mathbf{H}(t) = \mathbf{H}_1 \mathbf{H}^t = \mathbf{H}_1 \cdot \exp(t \cdot \log(\mathbf{H})) ,$$

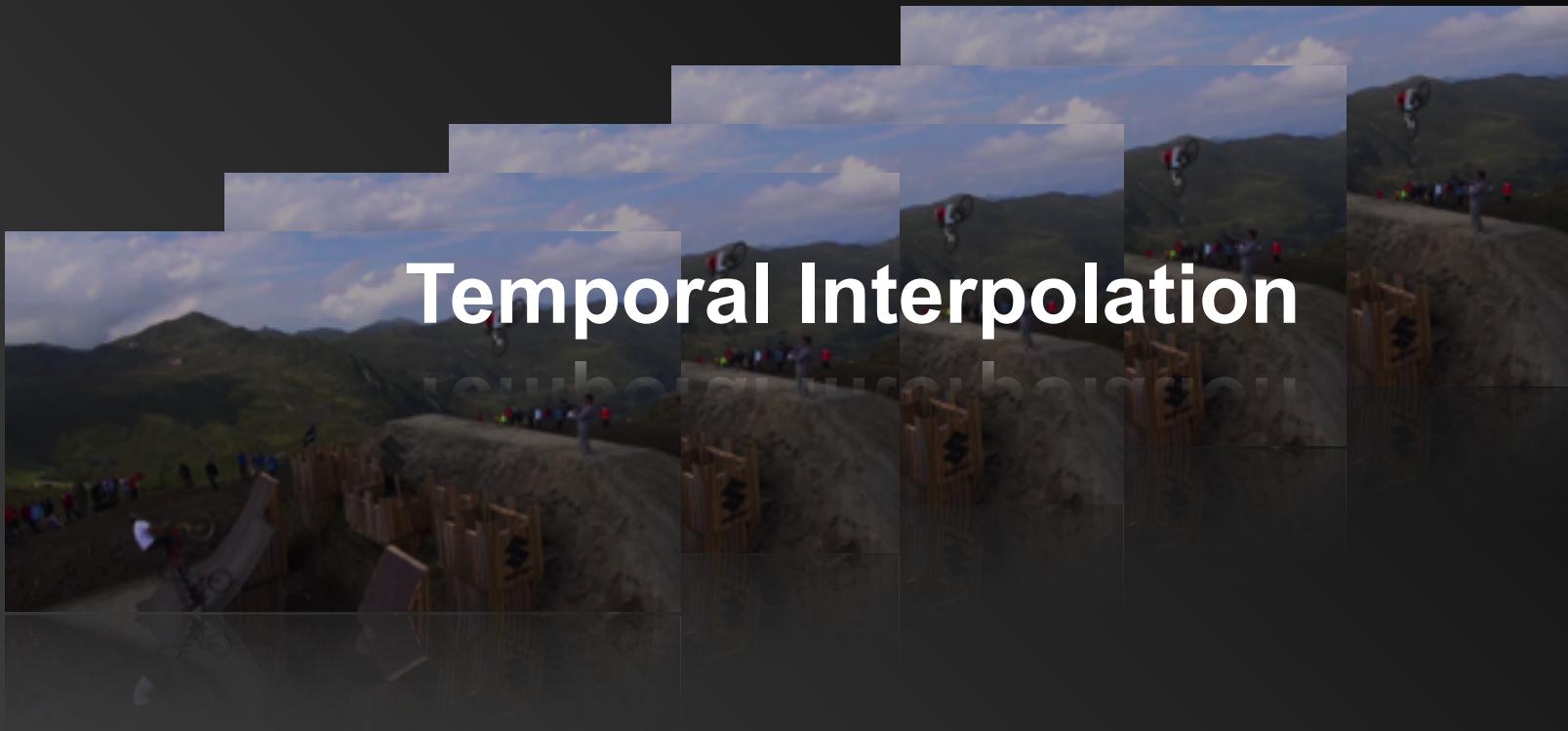
where

$$\mathbf{H} = \mathbf{H}_1^{-1} \mathbf{H}_2 .$$

$$\mathbf{H} = \mathbf{H}^{\mathbb{T}} - \mathbb{I} \mathbf{H}^{\mathbb{S}} .$$

# View Morphing





# Temporal Interpolation

- Optical flow to compute pixel-wise correspondence
- Transform images using backward warping
- Blend images using alpha-blending
- Implemented in fragment shader

# Experiments

# Experiments

- Middlebury Stereo dataset for ground truth RMSE
- Comparison to state-of-the-art results (e.g. Ballan et al., Lipski et al.)
- Comparison between our sparse approach and dense methods (e.g. Mahajan et al., Werlberger et al.)
- A lot of experiments, both on standard and self-made datasets (available online)

# Experiments



# Experiments



**Photo Exploration**

# Experiments



# Experiments



**Spatiotemporal Interpolation**

# Experiments



Virtual Avatar

# Conclusion

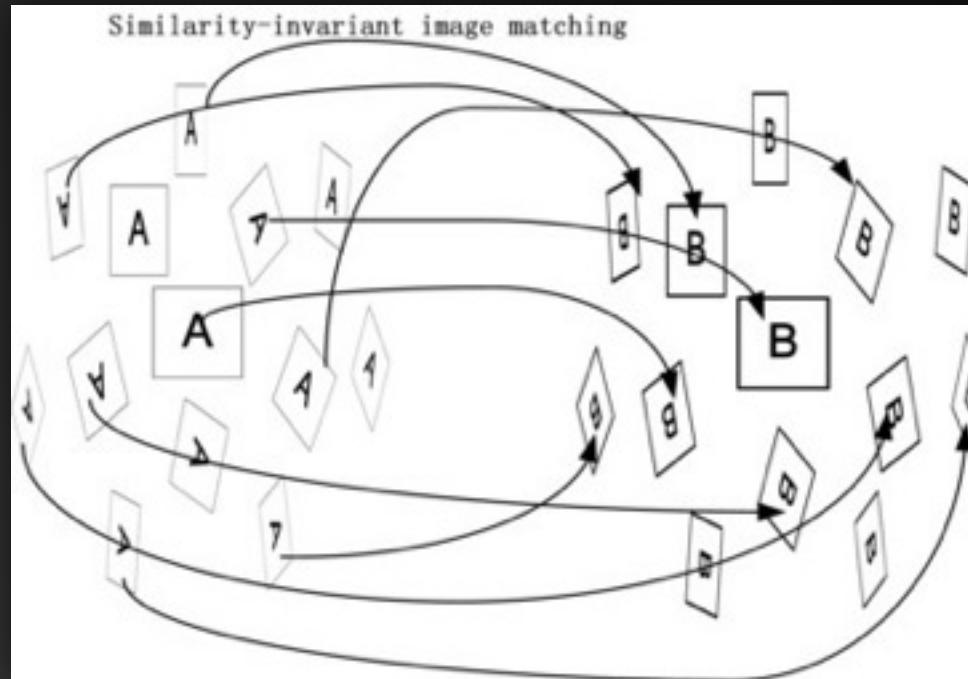
# Conclusion

- Presented autonomous, easy to use multi-view image sequence interpolation framework
- Introduced a suitable sparse feature detection, matching and filtering pipeline
- Proposed extrapolation scheme for missing border information
- Convincing results in various applications from super slow motion to virtual avatars
- Fairly easy to implement
- Binary and examples at [www.tobiasgurdan.de/research/](http://www.tobiasgurdan.de/research/)

# Thank you!

# Matcher

- ASIFT
  - Extends SIFT descriptor by missing affine parameters



# Matcher

- ASIFT
  - Extends SIFT descriptor by missing affine parameters
  - Sample parameter space and simulate affine transformations
  - Filter, filter, filter (identical, ambiguous, inlier matches)

# Matcher

- ASIFT
  - Source and target image



# Matcher

- ASIFT
  - Robust and Epipolar Matcher results

